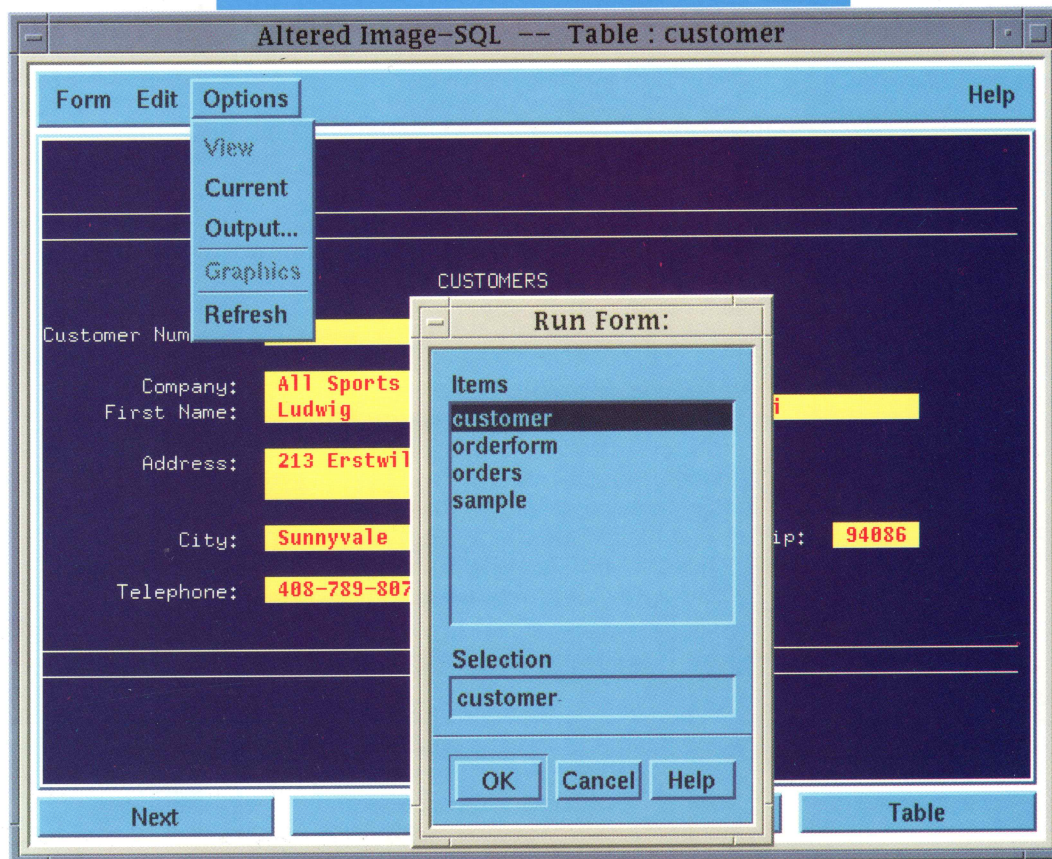


Deskterm™



**Bring GUI Ease of Use
to Existing Software**

The vast majority of UNIX[®] applications are character-based and unable to take advantage of user friendly graphical user interfaces (GUIs) such as Motif[™] and Windows 3.0[™]..... until now.

IXI's Deskterm[™] encapsulation tools allow you to quickly and easily add a GUI to a character application, bringing the benefits of a GUI, its low training requirement and increased productivity, to existing software. Your software can now run in a windowed, mouse-driven graphical environment featuring pull-down menus, scroll bars, pushbuttons, dialog boxes, resizable windows, cut and paste and multiple fonts and colors.



No X Knowledge Required

You don't need to know anything about X Window System[™] programming (at either Xlib or widget level) to move your application to Motif. At the heart of Deskterm is a powerful character protocol which implements traditional character terminal functionality and extends the capabilities to include GUI concepts such as menus, dialogs and pushbuttons.

There are two ways for you to use Deskterm: your application can be augmented to make direct use of the advanced features of the protocol; or you can use Deskterm Soft Option[™] to add the interface without changing the application at all. Typically the effort involved in adding a GUI is less than 10% of the original programming work. Much faster and less costly than redesigning your programs to work directly with X.

A major benefit of the Deskterm approach is that it exploits the existing character-based programming skills in your organization and allows you to concentrate your time and money on building applications and solutions rather than wrestling with X and the Motif toolkit.

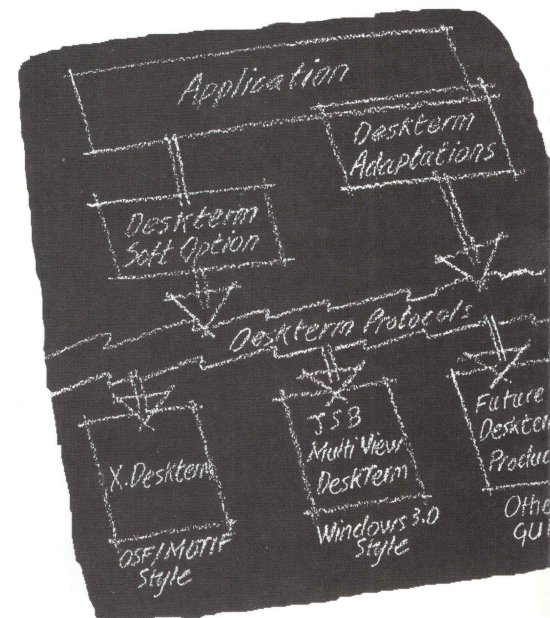
The Deskterm Protocol

The powerful character protocol used within Deskterm includes functionality you would expect from a character terminal. Operations such as cursor motion, character insertion and deletion, line and screen clearing and scrolling are implemented in the normal way.

A number of new options such as multiple fonts and colors have been added which are not present on many terminals. These, combined with line drawing and shading characters, allow clear and informative displays to be presented to the user. Screen resizing is also supported and the size of screen emulated can be changed in response to an explicit command from the application. Deskterm's design prevents naive applications having their display corrupted by an unexpected change in screen size.

Graphical user interface elements supported by the protocol include pull down menus, scrollbars, pushbuttons, popup dialogs and mouse interaction. The protocol provides commands to use and configure these elements. To reduce the effort of building an interface from scratch each time, a skeleton interface is provided by Deskterm, and the developer adjusts this template as required.

Just as a traditional terminal associates character sequences with special keys such as function and cursor keys, the Deskterm Protocol defines special sequences which are sent to the application to indicate menu selections, a response to a dialog, mouse clicks and drags and the like.



Deskterm Soft Option

Deskterm Soft Option allows a graphical interface to be constructed around an existing application. The new interface works by monitoring the display which is produced by the application, and translating user interaction with the display into commands understood by the application.

The graphical interface is described in an English-like script language which is designed to be easy to learn and use for all programmers. The Deskterm Soft Option language uses concepts

common to many programming languages and adds a number of facilities specially designed for the task of modelling an application and creating a new user interface.

Deskterm Soft Option translates the application output into the Deskterm Protocol. The developer does not need to know or understand the protocol to use Deskterm Soft Option, and the interface script need not include any references to the protocol.

Using Soft Option you can add value to your product by adding features not supported within the application code. For example, a typical database report generator will output the report continuously, making browsing of the report impossible without hard copy. With Soft Option, the output can be captured, and the user can then scroll back and forth through the text with a scroll bar. It is possible to give different users of a package a different interface, either according to their experience level, or to prevent access to parts of the application which they do not need to use.

Deskterm Soft Option Concepts

When running with Deskterm Soft Option, the application sees an ANSI or VT100 terminal interface, ensuring maximum compatibility with your existing character applications.

When a user sits at a terminal and uses an application, it appears as a set of screens, each of which performs some different function. The idea of a screen is also used within Deskterm Soft Option, and the first stage in building an interface is to identify the screens involved.

A *screen* provides the interface with context, and narrows down the options available at any one time. This has the advantage that the developer is not trying to build an interface which models everything that the application can ever do at any time - an impossible task for all but the simplest of programs. In addition, the most commonly used screens can be worked on first, leaving the rest of the application to be completed as time

allows. This means you can produce a demonstrable product for end user trials even faster.

Each screen is given a name by the developer. This allows the screen to be referenced from other parts of the interface so that the interface can switch screens when the application changes state.

Within each screen, Deskterm monitors different parts of the display both for application output and, for example, mouse clicks by the user. If a program displays a prompt on the bottom line of the screen, an interface could use this prompt to trigger the display of a dialog to request the information from the user. The original prompt can be hidden from the user in this case.

Soft Option allows the developer to say which parts of the display are "interesting" by defining *regions*. A region is an invisible rectangular part of the display and is defined in the following way:

REGION statusline IS AT
(x-origin, y-origin)
SIZE (width, height)

This statement sets up a region called *statusline* and gives the origin and size of the region.

Soft Option supports variables of integer, string and Boolean types. Named constants can be defined to avoid the use of arbitrary numbers in scripts. A library of over 200 predefined functions is provided, and user defined functions are also allowed.

The operation of the interface is coded as a set of *conditions* and *actions*. This structure allows the mapping of the event driven nature of the graphical interface to the command driven character interface. A typical condition to detect a particular pattern in a region could be

WHEN statusline CONTAINS
"Help"

This condition would be triggered whenever the word "Help" was

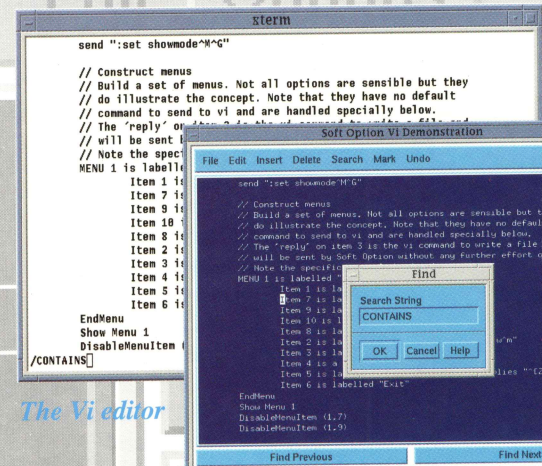
displayed *anywhere* in the statusline region. If the string was always displayed at the start of the region, the following condition could be used

WHEN statusline STARTS WITH
"Help"

In addition, the string "Help" could be replaced with a pattern defined as a regular expression. For example, the display of a number in the status line could be detected by

WHEN statusline CONTAINS
"[0-9][0-9]"*

The pattern in this case searches for any digit followed by zero or more digits followed by a space - in other words, a number of one or more digits. The ability to use patterns in this way gives the developer a very powerful way of tracking the state of the application and using the information displayed by the application to trigger changes in the interface presented to the user.



The Vi editor

The Soft Option version of Vi

The Soft Option Developer

To develop a Soft Option interface you only need to be a user of the application, you don't have to know its internals. A good graphical interface will make full use of the command set of the underlying application, so you need some familiarity with the displays produced by the application and the general way in which the screens are used.

A powerful development tool called *assist* is provided as part of the Deskterm Developer's Kit. This allows an application to be run, and screens and regions identified and recorded. Application output can be slowed down over the whole or just part of the screen to allow easier evaluation of the conditions necessary to trigger interface changes. For example, you may need to find a condition which can be taken to mean that the application has finished drawing its display, so that information can then be extracted from the screen. The ability to slow down the application so that the display updates can be monitored is very useful in these circumstances.

Items such as pushbuttons and pull down menus can be incorporated into your application very easily. The menus and buttons are set up using simple protocol sequences as part of your display initialisation. These sequences can be programmed in as predefined strings, or generated dynamically by your code. Often the simplest way to deal with the responses seen in the input stream by the application when menu items or buttons are selected is to treat them as if they were function keys. Each menu item and button generates a distinct response when selected, so the first button can be treated as key F1, the second as F2 and so on. Usually this involves no more than setting up the correct terminal configuration parameters.

Menu items and buttons can also be programmed to generate strings of characters when pressed, rather than individual responses. This extra configurability allows Deskterm to be configured to suit your application, rather than forcing you to work the other way around.

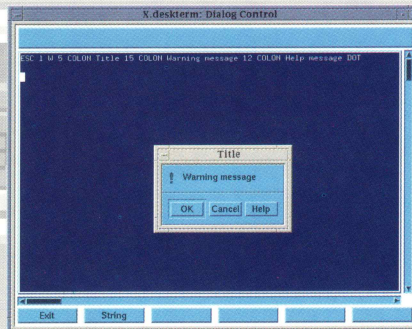
Using pop-up dialogs from your application is equally straightforward. A single protocol sequence contains both the command to start the dialog, and the parameters required, such as the help information to be displayed if the user clicks on the Help button. Deskterm dialogs are application modal - the user must reply to the dialog while it is displayed. The application knows that, having started the dialog, the only responses it expects from Deskterm are a dialog cancelled sequence or a dialog result sequence. This matches the normal command driven flow of control within most character applications.

IXI used the Deskterm Protocol to build a powerful hypertext help system - X.deskhelp[™]. Developed as an "open" help system, X.deskhelp allows manual pages to be displayed and browsed. The user can follow links from one page to another via menus, push buttons or mouse clicks on marked phrases. X.deskhelp uses the Deskterm Protocol to display bitmaps and pictures included in the help files to ease understanding of new information.

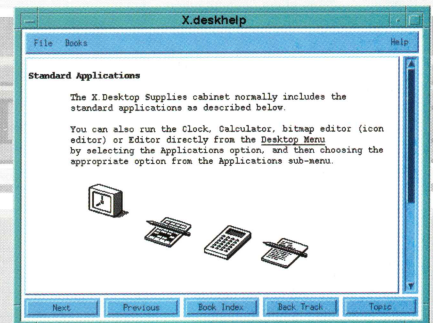
Using the Deskterm Protocol

To use the Deskterm Protocol directly from within your application, the first step is to configure the application to use the Deskterm sequences for cursor motion and other simple operations. This is often done using the Unix termcap or terminfo mechanism, but many developers have their own terminal configuration systems. Included in the Deskterm Developer's Kit are sample termcap and terminfo definitions for the protocol. These can be used as a guide and can be configured to meet your own requirements.

In some circumstances it may be more appropriate to use the Deskterm Protocol directly from your application by changing the input and output modules of your code. The output modules need to generate new sequences to set up menus, dialogs and so on, while the input modules need to be able to recognise the new sequences which may appear in the input stream from the user.



This illustrates how simple it is to generate a warning dialog box



The Deskterm Protocol contains all of the functionality required for the help interface. The development of the help system as a character program was faster than an equivalent Motif application and gave performance benefits. In addition, the use of Deskterm provides a high level of GUI independence which would otherwise be impossible to achieve.

CUSTOMERS

Run Form:

GUI Independence

Using Deskterm to build a GUI means that your application can run under Motif and Windows 3.0. The GUI is handled entirely through the Deskterm system, so switching from one user interface style to another does not require any changes to the application at all.

As a developer, you have the assurance that Deskterm will remain completely portable between different graphical environments. You can build an interface which will function equally effectively on X terminals and PC screens. In addition, your application will still work as expected by character terminal users - a big benefit for support and administration in installations with a mixture of character terminals, PCs and X terminals.

Reduce Time to Market

Deskterm reduces programming effort and time to market for a GUI version of your software. This is proven by the experiences of Uniplex, the leading supplier of office automation software. Uniplex calculated that it would have taken 40 man years to re-engineer their suite of software to work directly with X. Using the Deskterm approach, it took them only 2 man years. Not only did Uniplex reduce the development costs for a Motif version of the product but they brought their first GUI product to market years ahead of their competition.

Non-Unix Applications

Since all that Deskterm requires to operate is a character stream, any source of character data is acceptable

you to produce a unified user interface to applications running in mixed environments, so you can maintain your investment in existing applications.

Performance

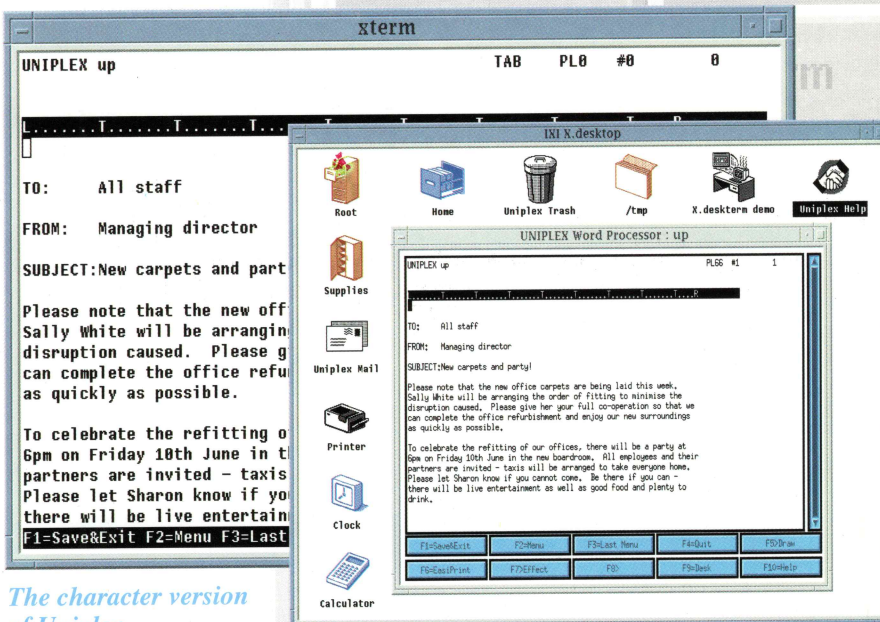
An important consideration for any developer is performance. The performance of an application using Deskterm compares favourably to an equivalent Motif port of the same application. The overall memory requirements will be similar and can often be smaller than running a native Motif application.

Deskterm is implemented as a separate process to your application. This architecture brings advantages in terms of GUI independence and the ability to use non-Unix applications. Splitting user interaction and X "exposure" handling into a separate process which is independent of your main application can make significant performance gains and memory savings which more than compensate for the small additional load passing characters back and forward.

Desktop Integration

The Deskterm Protocol takes advantage of the open interface offered by IXI's X.desktop™, the most widely used Motif compliant desktop manager. This allows character applications to control the user's desktop environment giving enhanced usability.

A major US corporation was able to add an integrated fax facility to a third party application. They were quoted \$10,000 by the authors of the application to add a fax option. Using Deskterm, the information to be faxed was captured from the display, formatted and stored in a text file by a Deskterm Soft Option script. A standalone fax program was then invoked via X.desktop with the text file as input. A "Fax" menu item was included in the new user interface, adding significant new functionality with minimal effort or disruption and very low cost.



The character version of Uniplex

Uniplex Windows, the Motif version of Uniplex

and can have a GUI added. The capabilities of modern networking mean that it is possible to use Deskterm with non-Unix applications. Deskterm is already being used to add graphical interfaces to Pick, VMS and MVS applications.

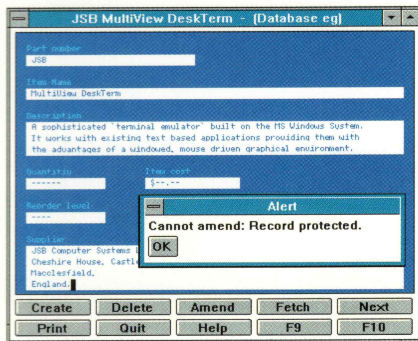
Deskterm can also be used to add a graphical front-end to software written in languages which do not have an X library (such as COBOL or FORTRAN). This flexibility allows

Telephone: 408-789-807

Supporting Motif and Windows 3.0

X.desktop™ is the X Windows and Motif front-end for the Deskterm system. It produces the X protocol messages an X display recognizes and adds the Motif look and feel using the official Motif toolkit.

The Deskterm approach also supports PC users. IXI licenses the Deskterm Protocol to JSB Computer Systems who have produced JSB MultiView DeskTerm™. This runs on DOS-based PCs and supports a Windows 3.0 interface to "desktermed" character-based applications running on a networked UNIX host.



Deskterm Developer's Kit

The Deskterm Developer's Kit contains everything you need to add a Motif GUI to your character-based application. Included is the Deskterm Protocol, X.desktop, Deskterm Soft Option and full documentation. IXI offers specialist training on the Deskterm software tools for those who need to get the software up and running fast.



The Deskterm Developer's Kit is available on the following platforms:

IBM RISC/System 6000

Sun 4, SPARCstations and SPARC clones

Ultron DECstations

HP9000/300, /400, /700, /800

SCO Open Desktop

NCR 3000

Data General AViiON family

Silicon Graphics

MIPS workstations

Call IXI for availability on other systems.

Wide recognition

Deskterm offers a flexible way of adding a GUI to your character-based applications. Since its launch in May 1990, Deskterm has received widespread acclaim including the accolade of UnixWorld's "Product of the Year 1990".

Deskterm has already been used to produce Motif versions of major UNIX applications: Uniplex Windows™ office automation software from Uniplex and a Motif and Windows 3.0 version of Informix-SQL database software from Altered Image. Major corporate end users are already reaping the benefits of the Deskterm approach that provides a quick and easy route into the world of graphical user interfaces.

About IXI

IXI is a leading developer of X Window System based products and services which helps users exploit the power of their UNIX based systems. Deskterm and X.desktop are the company's main products. Deskterm was introduced in May 1990 after two years of collaboration with major corporate end users and software vendors to develop a technology that met the broad range of needs in the marketplace.

X.desktop is an icon-based graphical user environment that provides an intuitive way of running programs, managing files and accessing printers and other peripherals. X.desktop is the world's most widely used UNIX desktop manager and is bundled with the Motif graphical user interfaces from IBM, the Santa Cruz Operation, NCR, Digital, NEC and many others.

X.desktop users can add the Deskworks productivity tools which have been carefully refined to meet popular activities of the professional user. The tools cover text editing, personal time management, mail and cross-network note facility. IXI's Motif Development Kit for Sun satisfies the need for the Motif development tools on a platform where the vendor does not support Motif.

IXI was founded in 1987 and has a strong global presence. A private company, headquartered in the European Community, IXI's financial backers include Save Roessler Bank AG, Summit Technology Fund and Provident.

Europe
IXI Limited
Vision Park
Cambridge CB4 4ZR
England

Tel: +44 223 236555
Fax: +44 223 236550

North America
IXI Corporation
One Annabel Lane
San Ramon
CA 94583

Tel: (510) 275 3120
Fax: (510) 275 0313

Japan
IXI Japan
1-1 Uchisaiwai-Cho 2 Chome
Chiyoda-Ku
Tokyo 100

Tel: +81 3 3 506 3471
Fax: +81 3 3 506 3499

Deskterm Soft Option includes technology licensed from Cambridge Connectivity Limited. X.desktop, Deskterm, X.Deskterm, Deskterm Soft Option and X.desktop are trademarks or registered trademarks of IXI Limited. JSB MultiView DeskTerm is a trademark of JSB Computer Systems Limited. Uniplex Windows is a trademark of Redwood International Limited. Informix is a registered trademark of Informix Software Inc. Altered Image is a trademark of Altered Image Software Limited. Motif is a trademark of the Open Software Foundation. Windows 3.0 is a trademark of Microsoft. All other trademarks are acknowledged.